

How Signsoft intelliBC4 runs with the SAP Web Application Server 6.40

intelliBC4

How Signsoft intelliBO 4 runs with the SAP Web Application Server 6.40

Copyright Information

This publication is protected by German and international copyright law. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of Signsoft.

© 1998 - 2007 Signsoft GmbH

All Rights reserved.

Signsoft - Softwareentwicklungs- und -vertriebsgesellschaft mbH

Leipziger Str. 118, 01127 Dresden, Germany

<http://www.signsoft.com>, <http://www.intellibo.com>, <http://www.intellixo.com>

E-Mail: info@signsoft.com

Brand and Product Names

All brand and product names are trademarks or registered trademarks of their respective owners.

Version: 1.1
Author: Christian Keiler
Last change: Frank Schwarz (2007-01-15)

History:

Christian Keiler (2006-01-07) Version 1.0
Initial release
Frank Schwarz (2007-01-11) Version 1.1
Update for SAP NetWeaver 2004s SR2 (SP9)

Table of Contents

1 Abstract	4
2 Issue	4
3 Signsoft intelliBO	4
4 Deployment of the Signsoft intelliBO JCA adapter.....	5
5 Conclusion	9

1 Abstract

This paper describes how Signsoft intelliBO 4 can be used with the SAP Web Application Server 6.40. At first there is given a short introduction of the Signsoft intelliBO JCA¹ Adapter and its use. Secondly an example from the Signsoft intelliBO distribution is utilized to demonstrate the seamless integration of Signsoft intelliBO into the SAP Web Application Server.

2 Issue

Requiring transparent persistence of Java object models in transactional data stores, Java Data Objects (JDO) will sensibly be used. Regarding this issue an easy-to-use integrated development environment (IDE) might also be preferred, which provides the possibility of defining an appropriate mapping between plain Java objects and database tables as Signsoft intelliBO does.

The SAP Web Application Server is widely utilized to shift from monolithic ERP² applications to modular J2EE applications. At the same time, new persistence strategies for enterprise data are to be devised. Signsoft intelliBO will smoothen the transition.

3 Signsoft intelliBO

Signsoft intelliBO is an implementation of the Java Data Objects Specification, defined by the JSR 12. Using this specification as a good starting point, intelliBO provides a full-fledged O/R mapper to conveniently connect the object-oriented with the entity-relationship view on enterprise data. Even prominent features of the forthcoming JDO 2.0 specification, as an attach/detach mechanism for persistent objects, are already available. Signsoft intelliBO also includes a JCA adapter with support for Local-Transactions and XA-Transactions, allowing an easy integration with the SAP Web Application Server.

The Java Connector Architecture defines an interface contract between different systems, for instance an application server and a JDO implementation. It consists of the following components:

- System level contracts, concerned for instance with transaction handling, security and connection management
- Deployment descriptions, which hold the definition of the content of a connector and its configuration options
- Common Client Interface (CCI), which is a client-side API for collaboration with different Enterprise Information Systems (EIS)

Integration via JCA keeps applications portable, because all server-specific settings can be configured by editing simple XML deployment descriptors. The

1 Java Connector Architecture (JSR016), <http://java.sun.com/j2ee/connector/>

2 Enterprise Resource Planning

How Signsoft intelliBO 4 runs with the SAP Web Application Server 6.40

JCA adapter provides connectivity between the application server and intelliBO. Using JCA to integrate Signsoft intelliBO into the application server enables enlistment in application server transactions and application-server-managed pooling of PersistenceManager handles. Transaction demarcation will not have to be handled in the application code, instead the application server will manage the transaction (container-managed transactions).

Signsoft intelliBO ships with an Ant script that automatically creates a resource archive (.rar) comprising all the jars and configuration files necessary for the JCA adapter.

The Signsoft intelliBO adapter and runtime system consists of the following libraries and configuration files:

- *ibo.jar*,
- *iboaux.jar*,
- *license.xml*,
- *ibo-config.xml*,
- *ra.xml*,
- an application server specific configuration file for the adapter, and
- the JDBC database driver

At deployment time, the connector will be bound into the application servers JNDI tree under a name specified by the application developer. This name can be used from application components to obtain a connection to the resources of Signsoft intelliBO.

4 Deployment of the Signsoft intelliBO JCA adapter

Instead of a self-managed JDBC connection, pool an application may also use a data source configured in the application server. In this case, only the JNDI name of the data source has to be configured in *ibo-config.xml*:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE node SYSTEM "storage.dtd">
<node name="ibo-config">
  <entry key="version" value="1.0" />
  <node name="environment">
    <entry key="optimistic" value="true" />
    <entry key="logging" value="false" />
    <entry key="non-transactional-read" value="true" />
    <entry key="non-transactional-write" value="false" />
    <entry key="use-shared-ntx" value="true" />
  </node>
</node>
```

How Signsoft intelliBO 4 runs with the SAP Web Application Server 6.40

```
</node>
<node name="resource-entry">
  <entry key="type" value="ds"/>
  <entry key="name" value="SJDO_DEFAULT"/>
  <entry key="jndi-name" value="JndiDS"/>
  <entry key="dbsupport-name"
    value=
    "com.signsoft.ibo.dbsupport.oracle81x.OracleDatabaseInfo"/>
</node>
</node>
```

Please note: It is necessary to let Signsoft intelliBO manage the transaction branch of this connection. Therefore, the data source must not take part in the application server's transaction directly, but it will have to be placed under the control of Signsoft intelliBO's transaction manager.

Transaction-related code does not occur in the application components using JCA for integrating Signsoft intelliBO with the SAP Web Application Server. Signsoft intelliBO will automatically enlist and take part in the application server's transactions according to the settings specified in the application components deployment descriptors.

Signsoft intelliBO supports both local transactions and distributed (XA) transactions by implementing the two-phase commit (2-PC) protocol. This allows cooperation with other XA compliant resources (for instance a messaging system) within a transaction. With the current version of the SAP Web Application Server, both local and XA transactions can be applied.

In addition to the PersistenceManager cache specified by the JDO standard, Signsoft intelliBO features a second cache level, which is shared between all PersistenceManagers of a Java Virtual Machine. In a clustered environment this cache should be either disabled or the intelliBO Distributed Cache should be used. More information about the configuration settings and the use of the Distributed Cache can be found in the Distributed Cache Userguide shipped with Signsoft intelliBO.

Deployment of applications should be done separately for each module, such as the web module (WAR), the bean's module (JAR) and the JCA module (RAR).

For creating a deployable intelliBO JCA connector the build system under *IBO_HOME/samples/architecture/j2ee* from the intelliBO distribution can be used. This example provides a good starting point for creating a build system for real applications.

Deployment descriptor of the intelliBO connector (*ra.xml*):

How Signsoft intelliBO 4 runs with the SAP Web Application Server 6.40

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE connector PUBLIC
  "-//Sun Microsystems, Inc.//DTD Connector 1.0//EN"
  "http://java.sun.com/dtd/connector_1_0.dtd">
<connector>
  <display-name>
    Signsoft_intelliBO_JDO_Persistence_Provider
  </display-name>
  <vendor-name>Signsoft GmbH</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>JDO RDBMS</eis-type>
  <version>1.0.0</version>
  <resourceadapter>
    <managedconnectionfactory-class>
com.signsoft.ibo.client.jca.ManagedPersistenceManagerFactory
    </managedconnectionfactory-class>
    <connectionfactory-interface>
com.signsoft.ibo.client.jca.IBOJCAPersistenceManagerFactory
    </connectionfactory-interface>
    <connectionfactory-impl-class>
com.signsoft.ibo.client.jca.JCAPersistenceManagerFactory
    </connectionfactory-impl-class>
    <connection-interface>
com.signsoft.ibo.client.jca.IBOJCAPersistenceManager
    </connection-interface>
    <connection-impl-class>
com.signsoft.ibo.client.jca.JCAPersistenceManager
    </connection-impl-class>
    <transaction-support>
      XATransaction
    </transaction-support>
    <config-property>
      <config-property-name>
        LocalTXEvents
      </config-property-name>
    </config-property>
  </resourceadapter>
</connector>
```

```
<config-property-type>
    java.lang.Boolean
</config-property-type>
<config-property-value>
    false
</config-property-value>
</config-property>
<reauthentication-support>
    false
</reauthentication-support>
</resourceadapter>
</connector>
```

The given example from Signsoft intelliBO in *IBO_HOME/samples/architecture/j2ee* can now be deployed in the SAP Web Application Server as follows:

1. First, build the application with Ant. This can be achieved by calling

ant ejb war rar

from the command line in directory *IBO_HOME/samples/architecture/j2ee*. Please ensure that all needed libraries, i.e. JDBC drivers, are located in the *lib/* directory of the sample. Please check that the *ra.xml* looks like the above-stated one. Secondly, call ***DeployTool.bat*** from the SAP Web AS distribution. With this tool, you will prepare the enterprise application for its deployment. Please create a new project within this tool and go to the ***Assembler*** tab. By using the shortcut ***Ctrl + D (menu item Assemble -> Add Archive)***, you add each archive from the *dist/* directory of the sample, i.e. *beans.jar*, *ibo.rar*, and *web.war*.

2. The next step is to import the deployment descriptors generated by Ant. For this, use the ***J2EE Components*** tab. Following the Signsoft intelliBO example, add an ***EJB Group***. This can be achieved by choosing the ***Load Existing*** radio button in the upcoming ***Load EJB JAR*** window and set the path to the appropriate *ejb-jar.xml* file (*dist/ejb/META-INF/ejb-jar.xml* in the samples directory). Furthermore, the ***Files Directory*** has to be set to *IBO_HOME/samples/architecture/j2ee/dist/ejb*, too! The obligatory field ***Archive Name*** has to be set to "beans".
3. Add the *web.xml* from *dist/war/WEB-INF* using ***Add Web*** and set the Files Directory to *IBO_HOME/samples/architecture/j2ee/dist/war*. Let "web" be the ***Archive Name***.
4. The JCA adapter does not need any special treatment. Adding the file

ibo.rar in the **Assembler tab** as you did in step 1 is just enough for the deployment preparation.

5. After that, you can make all archives using the shortcut **Ctrl+M** (menu item **J2EE Components -> Make All Achives**). Since you still need the EAR file, go to the **Assembler** tab again and use the shortcut **Ctrl+M** (menu item **Assemble -> Make Ear**), name it "**ibo.ear**".
6. The complete enterprise archive can be found in the **Deployer** tab. Finally, you should check if all JNDI names are correctly set. The JNDI name for the Signsoft intelliBO resource adapter has to be *eis/ibo*³. This setting can be found by selecting the *ibo.rar* component and choosing the tab **Server Settings -> Additional Settings -> JNDI Name**. For the enterprise java bean, *PersonManager* the JNDI name must be "*PersonManager*". This setting can be found by selecting the entry **PersonManager** in beans.jar and going to the tab **General**.
7. The web context of the application must be set to "*intellibo*". Please select the entry *ibo.ear*, tab **Descriptor**, subtab **Context, WAR Files web.war -> Context Root "intellibo" (Modify)**.

Now connect to the SAP Web AS using the menu item **Deploy -> Connect** and deploy the example application with the shortcut **Ctrl+E** (menu item **Deploy -> Deployment -> Deploy Ear**) and start the application by answering **OK** in the upcoming dialog right after the deployment is done. After deploying and starting, you can find the example application via **SAP Visual Administrator (Server -> Services -> Deploy -> sap.com/<your project>)** or via a web browser (*http://localhost:50000/intellibo/* might be a good guess).

5 Conclusion

This white paper has shown how to integrate Signsoft intelliBO 4 into SAP Web Application Server 6.40. Starting with a straightforward example, you can use the same procedure to deploy more complex applications to the SAP Web Application Server. Since integration is manageable this way, you can benefit from the Signsoft intelliBO resource adapter in enterprise applications on the SAP Web Application Server.

³ This is a sample specific setting. You are free to use any JNDI compliant name as long as your deployment descriptors reflect this name unambiguously.